



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/688,640	10/17/2003	Antti Kokkinen	200701903-2	1958
22879 7590 07/21/2009 HEWLETT PACKARD COMPANY P O BOX 272400, 3404 E. HARMONY ROAD INTELLECTUAL PROPERTY ADMINISTRATION FORT COLLINS, CO 80527-2400				
EXAMINER				
WANG, BEN C				
ART UNIT		PAPER NUMBER		
2192				
NOTIFICATION DATE		DELIVERY MODE		
07/21/2009		ELECTRONIC		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM

ipa.mail@hp.com

jessica.l.fusek@hp.com

Office Action Summary

Application No.

10/688,640

Applicant(s)

KOKKINEN, ANTTI

Examiner

BEN C. WANG

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 30 March 2009.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-24 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-24 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SF/ICE)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

1. Applicant's amendment dated March 30, 2009, responding to the Non-Final Office action mailed December 30, 2008 provided in the rejection of claims 1-24.

Claims 1-24 remain pending in the application and which have been fully considered by the examiner.

Applicant's arguments with respect to claims rejection have been fully considered but are moot in view of the new grounds of rejection – see *O'Neil* and *Crudele et al.* - arts made of record, as applied hereto.

Claim Rejections – 35 USC § 103(a)

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Patrick J. O'Neil (WO 02/41147 A1) (hereinafter 'O'Neil' – art made of record)

3. **As to claim 1** (Previously Presented), O'Neil discloses a method for updating software in an electronic device, the method comprising:

- generating an update package for updating at least one software application being generated based upon difference information between the at least one

- software application and at least one reference software installed on the electronic device (e.g., Fig. 1A, elements 110 – Update Package; 102 – Update Generator; 106 – First Version; 108 – Second Version; P. 5, 1st non-full Para - ... identifies differences between of the updated operating code and the resident operating code and thereafter generates an update package ...; P. 5, last Para - ... an update generator that performs a version comparison between the first code version and the second code version to identify pattern differences ...; P. 12, 1st non-full Para - ... Alternatively, the update generator 102 may be equipped with the ability to generate and provide a plurality of update packages 110 ... the update generator 102 may create a version manifest (*reference software*), which comprises a list of archived update packages 110 ...; P. 15, 1st full Para - ... a server manifest comprising a list of archived update packages ... may transfer the server manifest to the one or more client devices ... then review the manifest and submit a request for the update package ... to be transferred from the update server array 122 to the one or more client devices ...);
- updating the at least one software application using the update package and the reference software (e.g., P. 20, last Para - ... the update installation process advances to a state 208 where the client device 104 compares the update signature in server manifest (*reference software*) with that of the existing operational software ... if the update package 104 should be ... installed on the client device 104 ...); and

- wherein the updating leaves the at least one reference software unchanged (e.g., P. 12, 1st non-full Para - ... the update generator 102 may create a version manifest (reference software), which comprises a list of archived update packages 110 ...; P. 20, 1st full Para - ... the polled server manifest (reference software) comprises information used to determine the latest available version of software)

4. **As to claim 2** (Original) (incorporating the rejection in claim 1), O'Neil discloses the method wherein generating an update package for updating the at least one software application based upon the at least one reference software installed on the electronic device comprises:

- accessing a copy of the at least one reference software (e.g., P. 20, last Para – Once the server manifest is transferred to and obtained by the client device 104 ...);
- retrieving a copy of the at least one software application (e.g., P. 20, 1st full Para - ... server manifest comprises information used to determine the latest available version of the software ...); and
- generating an update package (e.g., Fig. 1A, elements 110 – Update Package; 102 – Update Generator; 106 – First Version; 108 – Second Version; P. 5, 1st non-full Para - ... identifies differences between of the updated operating code and the resident operating code and thereafter generates an update package ...)

5. **As to claim 3** (Original) (incorporating the rejection in claim 1), O'Neil discloses the method wherein generating an update package for updating at least one software application based upon the at least one reference software installed on the electronic device comprises:

- accessing a copy of the at least one reference software (e.g., P. 20, last Para – Once the server manifest is transferred to and obtained by the client device 104 ...);
- retrieving a copy of each of multiple versions of the at least one software application (e.g., P. 20, 1st full Para - ... server manifest comprises information used to determine the latest available version of the software ...); and
- generating an update package comprising all transitions between the retrieved versions of the at least one software application (e.g., Fig. 1A, elements 110 – Update Package; 102 – Update Generator; 106 – First Version; 108 – Second Version; P. 5, 1st non-full Para - ... identifies differences between of the updated operating code and the resident operating code and thereafter generates an update package ...)

6. **As to claim 4** (Original) (incorporating the rejection in claim 1), O'Neil discloses the method further comprising updating multiple update versions of the at least one software application installed on the electronic device is performed using a single update package (e.g., P. 22, 1st full Para - ... proceed to update the file or component

using the most current version available without sequentially applying each possible update package (i.e. version 1 updated directly to version 3) ...)

7. **As to claim 5** (Original) (incorporating the rejection in claim 1), O'Neil discloses the method further comprising installing the at least one software application and the at least one reference software as part of a single installation (e.g., Fig. 1A, elements 110 – Update Package; 102 – Update Generator; 106 – First Version; 108 – Second Version; P. 5, 1st non-full Para - ... identifies differences between of the updated operating code and the resident operating code and thereafter generates an update package ...; P. 5, last Para - ... an update generator that performs a version comparison between the first code version and the second code version to identify pattern differences ...; P. 12, 1st non-full Para - ... Alternatively, the update generator 102 may be equipped with the ability to generate and provide a plurality of update packages 110 ... the update generator 102 may create a version manifest (reference software), which comprises a list of archived update packages 110 ...; P. 15, 1st full Para - ... a server manifest comprising a list of archived update packages ... may transfer the server manifest to the one or more client devices ... then review the manifest and submit a request for the update package ... to be transferred from the update server array 122 to the one or more client devices ...)

8. **As to claim 6** (Original) (incorporating the rejection in claim 1), O'Neil discloses the method further comprising updating the at least one reference software and

updating the at least one software application as part of a single update (e.g., Fig. 1A, elements 110 – Update Package; 102 – Update Generator; 106 – First Version; 108 – Second Version; P. 5, 1st non-full Para - ... identifies differences between of the updated operating code and the resident operating code and thereafter generates an update package ...; P. 5, last Para - ... an update generator that performs a version comparison between the first code version and the second code version to identify pattern differences ...; P. 12, 1st non-full Para - ... Alternatively, the update generator 102 may be equipped with the ability to generate and provide a plurality of update packages 110 ... the update generator 102 may create a version manifest (*reference software*), which comprises a list of archived update packages 110 ...; P. 15, 1st full Para - ... a server manifest comprising a list of archived update packages ... may transfer the server manifest to the one or more client devices ... then review the manifest and submit a request for the update package ... to be transferred from the update server array 122 to the one or more client devices ...)

9. **As to claim 7** (Original) (incorporating the rejection in claim 1), O'Neil discloses the method wherein the at least one software application comprises a plurality of software applications, and the at least one reference software comprises a plurality of reference software (e.g., Fig. 1A, elements 110 – Update Package; 102 – Update Generator; 106 – First Version; 108 – Second Version; P. 5, 1st non-full Para - ... identifies differences between of the updated operating code and the resident operating code and thereafter generates an update package ...; P. 5, last Para - ... an update

generator that performs a version comparison between the first code version and the second code version to identify pattern differences ...; P. 12, 1st non-full Para - ...
Alternatively, the update generator 102 may be equipped with the ability to generate and provide a plurality of update packages 110 ... the update generator 102 may create a version manifest (reference software), which comprises a list of archived update packages 110 ...; P. 15, 1st full Para - ... a server manifest comprising a list of archived update packages ... may transfer the server manifest to the one or more client devices ... then review the manifest and submit a request for the update package ... to be transferred from the update server array 122 to the one or more client devices ...)

10. **As to claim 8** (Original) (incorporating the rejection in claim 7), O'Neil discloses the method further comprising:

- identifying a software application needing updating from the plurality of software applications installed on the electronic device (e.g., P. 12, 1st non-full Para - ...
Alternatively, the update generator 102 may be equipped with the ability to generate and provide a plurality of update packages 110 ... the update generator 102 may create a version manifest (reference software), which comprises a list of archived update packages 110 ...; P. 15, 1st full Para - ... a server manifest comprising a list of archived update packages ...; P. 20, 1st full Para - ... server manifest comprises information used to determine the latest available version of the software ...)

- identifying whether a reference software corresponding to the software application needing updating is present on the electronic device, wherein if the reference software is not present, then installing the software application and an associated reference software in a single update on the electronic device (e.g., P. 15, 1st full Para - ... a server manifest comprising a list of archived update packages ... may transfer the server manifest to the one or more client devices ... then review the manifest and submit a request for the update package ... to be transferred from the update server array 122 to the one or more client devices ...)

11. **As to claim 9** (Original) (incorporating the rejection in claim 7), O'Neil discloses the method further comprising:

- identifying a software application needing updating from the plurality of software applications installed on the electronic device (e.g., P. 20, last Para – Once the server manifest is transferred to and obtained by the client device 104 ...); and
- identifying whether a reference software corresponding to the software application needing updating is present on the electronic device, wherein if the reference software is present then retrieving an update package for the software application needing updating (e.g., P. 20, 1st full Para - ... server manifest comprises information used to determine the latest available version of the software ...); and
- installing the update package on the electronic device (e.g., P. 20, last Para - ... the update installation process advances to a state 208 where the client device

104 compares the update signature in server manifest (*reference software*) with that of the existing operational software ... if the update package 104 should be ... installed on the client device 104 ...)

12. **As to claim 10** (Original) (incorporating the rejection in claim 7), O'Neil discloses the method further comprising:

- identifying a software application needing updating from the plurality of software applications installed on the electronic device (e.g., P. 20, last Para - ... if the update package 104 should be downloaded from the update device server 136 and installed on the client device 104 ...);
- determining if the update is needed immediately (e.g., P. 13, last 2nd Para - ... Each client device 104 may then request transfer of the desired update package 110 which is selectively sent by the server ...); and
- storing the update until the update is needed immediately (e.g., P. 13, last 2nd Para - ... the update generator 102 generates and archives a plurality of update packages 110 for distribution to one or more different types of client devices 104 ..)

13. **As to claim 11** (Original) (incorporating the rejection in claim 10), O'Neil discloses the method wherein when the update is determined to be needed immediately, then

- invoking an update agent to employ at least the stored update package and reference software (e.g., P. 6, 1st non-full Para - ... an update agent ... transforming the first code version resident in the electronic device into the updated second code version ...); and
- updating the software application with the update package (e.g., P. 20, last Para - ... the update installation process advances to a state 208 where the client device 104 compares the update signature in server manifest (*reference software*) with that of the existing operational software ... if the update package 104 should be ... installed on the client device 104 ...)

14. **As to claim 12** (Previously Presented), O'Neil discloses a system for updating software, the system comprising:

- an electronic device capable of having software installed thereon (e.g., P. 17, 2nd full Para - ... the update management system may be used .. such as cellular or mobile phones ...);
- a software delivery device for receiving and installing a reference software to the electronic device if the electronic device does not have the reference software previously installed (e.g., P. 15, 1st full Para - ... a server manifest comprising a list of archived update packages ... may transfer the server manifest to the one or more client devices ... then review the manifest and submit a request for the update package ... to be transferred from the update server array 122 to the one or more client devices ...); and

- the software delivery device receiving and delivering at least one update package to the electronic device, wherein the at least one update package is based on differences between at least one application software and the reference software, and the reference software facilitates, using the at least one update package, at least one update to the application software installed on the electronic device, and wherein the updating leaves the reference software unchanged (e.g., Fig. 1A, elements 110 – Update Package; 102 – Update Generator; 106 – First Version; 108 – Second Version; P. 5, 1st non-full Para - ... identifies differences between of the updated operating code and the resident operating code and thereafter generates an update package ...; P. 5, last Para - ... an update generator that performs a version comparison between the first code version and the second code version to identify pattern differences ...; P. 12, 1st non-full Para - ... Alternatively, the update generator 102 may be equipped with the ability to generate and provide a plurality of update packages 110 ... the update generator 102 may create a version manifest (reference software), which comprises a list of archived update packages 110 ...; P. 15, 1st full Para - ... a server manifest comprising a list of archived update packages ... may transfer the server manifest to the one or more client devices ... then review the manifest and submit a request for the update package ... to be transferred from the update server array 122 to the one or more client devices ...; P. 12, 1st non-full Para - ... the update generator 102 may create a version manifest (reference software), which comprises a list of archived update packages 110 ...; P. 20, 1st full Para -

... the polled server manifest (*reference software*) comprises information used to determine the latest available version of software)

15. **As to claim 13** (Original) (incorporating the rejection in claim 12), O'Neil discloses the system wherein the electronic device further comprises an update agent, the update agent being capable of employing the reference software in conjunction with any retrieved update package to generate updated versions of the application software and also being capable of updating a plurality of application software employing reference software associated with each application software (e.g., P. 6, 1st non-full Para - ... an update agent ... transforming the first code version resident in the electronic device into the updated second code version ...; P. 20, last Para - ... the update installation process advances to a state 208 where the client device 104 compares the update signature in server manifest (*reference software*) with that of the existing operational software ... if the update package 104 should be ... installed on the client device 104 ...)

16. **As to claim 14** (Original) (incorporating the rejection in claim 12), O'Neil discloses the system further comprising an update generating system, the update generating system comprising a loader manager, the loader manager:

- managing loading of application software and application software version updates from the software delivery device (e.g., P. 20, last Para - ... the update installation process advances to a state 208 where the client device 104

compares the update signature in server manifest (*reference software*) with that of the existing operational software ... if the update package 104 should be ... installed on the client device 104 ...);

- employing a loader from a loader module (e.g., P. 43, last second full Para - ... the update agent 1025 may include functionality for performing client operations associated with update package management ...); and
- employing security services to authenticate software being delivered (e.g., P. 43, last second full Para - ... the update agent 1025 may include functionality used to prepare the update package 110 such ... encryption/un-encryption, and/or validation of the contents of the update package 110)

17. **As to claim 15** (Original) (incorporating the rejection in claim 14), O'Neil discloses the system wherein the loader manager further comprises an installation agent for installing application software and downloading files from the software delivery device (e.g., Fig. 8B; P. 42, last 2nd full Para - .. includes a download agent 1020 and an update agent 1025 used to process the update package 110 and perform the update functions ...; P. 43, last second full Para - ... the update agent 1025 may include functionality for performing client operations associated with update package management ...)

18. **As to claim 16** (Original) (incorporating the rejection in claim 16), O'Neil discloses the system wherein the loader manager is adapted to:

- identify an application software needing updating (e.g., P. 12, 1st non-full Para - ... Alternatively, the update generator 102 may be equipped with the ability to generate and provide a plurality of update packages 110 ... the update generator 102 may create a version manifest (reference software), which comprises a list of archived update packages 110 ...; P. 15, 1st full Para - ... a server manifest comprising a list of archived update packages ...; P. 20, 1st full Para - ... server manifest comprises information used to determine the latest available version of the software ...);
- identify whether reference software associated with the application software needing updating exists (e.g., P. 15, 1st full Para - ... a server manifest comprising a list of archived update packages ... may transfer the server manifest to the one or more client devices ... then review the manifest and submit a request for the update package ... to be transferred from the update server array 122 to the one or more client devices ...); and
- coordinating an update of the application software and an associated reference software in a single update (e.g., Fig. 8B; P. 42, last 2nd full Para - .. includes a download agent 1020 and an update agent 1025 used to process the update package 110 and perform the update functions ...; P. 43, last second full Para - ... the update agent 1025 may include functionality for performing client operations associated with update package management ...)

19. **As to claim 17** (Original) (incorporating the rejection in claim 14), O'Neil discloses the system wherein the loader manager is adapted to:

- retrieve the update package (e.g., P. 20, 1st full Para - ... server manifest comprises information used to determine the latest available version of the software ...);
- access contents of the update package (e.g., P. 43, last second full Para - ... the update agent 1025 may include functionality for performing client operations associated with update package management ...); and
- verify the update package (e.g., P. 43, last second full Para - ... the update agent 1025 may include functionality used to prepare the update package 110 such ... encryption/un-encryption, and/or validation of the contents of the update package 110)

20. **As to claim 18** (Original) (incorporating the rejection in claim 14), O'Neil discloses the system wherein the loader manager is adapted to determine immediacy of a needed update for a particular application software (e.g., P. 12, 1st non-full Para - ... Alternatively, the update generator 102 may be equipped with the ability to generate and provide a plurality of update packages 110 ... the update generator 102 may create a version manifest (reference software), which comprises a list of archived update packages 110 ...; P. 15, 1st full Para - ... a server manifest comprising a list of archived update packages ...; P. 20, 1st full Para - ... server manifest comprises information used to determine the latest available version of the software ...)

21. **As to claim 19** (Original) (incorporating the rejection in claim 12), O'Neil discloses the system wherein the software delivery device is one of a server, a CDROM, and a network (e.g., Fig. 1C, element 132 – Update Server Array; P. 19, last non-full Para - ... efficiency of the update distribution systems ... to distribute ... update packages ... wireless communications mediums, such as the Internet or wireless local area networks (WLAN))

22. **As to claim 20** (Original) (incorporating the rejection in claim 12), O'Neil discloses the system wherein the electronic device is one of a computer, a digital phone, and a digital camera (e.g., P. 17, 2nd full Para - ... the update management system may be used in conjunction with over the air updating of portable electronic device, such as cellular or mobile phones ...)

23. Claims 21-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over O'Neil in view of Crudele et al. (Pub. No. US 2002/0099726 A1) (hereinafter 'Crudele' – art made of record)

24. **As to claim 21** (Original), O'Neil discloses a method for updating software in an electronic device the method comprising:

- generating a first update package for updating at least one software application, the first update package being generated based upon difference information between first and second software versions (e.g., Fig. 1A, elements 110 –

Update Package; 102 – Update Generator; 106 – First Version; 108 – Second Version; P. 5, 1st non-full Para - ... identifies differences between of the updated operating code and the resident operating code and thereafter generates an update package ...; P. 5, last Para - ... an update generator that performs a version comparison between the first code version and the second code version to identify pattern differences ...; P. 12, 1st non-full Para - ... Alternatively, the update generator 102 may be equipped with the ability to generate and provide a plurality of update packages 110 ... the update generator 102 may create a version manifest (reference software), which comprises a list of archived update packages 110 ...; P. 15, 1st full Para - ... a server manifest comprising a list of archived update packages ... may transfer the server manifest to the one or more client devices ... then review the manifest and submit a request for the update package ... to be transferred from the update server array 122 to the one or more client devices ...); and

- generating a second update package for updating the at least one software application, the second update package being generated based upon difference information between first and third software versions (e.g., Fig. 1A, elements 110 – Update Package; 102 – Update Generator; 106 – First Version; 108 – Second Version; P. 5, 1st non-full Para - ... identifies differences between of the updated operating code and the resident operating code and thereafter generates an update package ...; P. 5, last Para - ... an update generator that performs a version comparison between the first code version and the second code version

to identify pattern differences ...; P. 12, 1st non-full Para - ... Alternatively, the update generator 102 may be equipped with the ability to generate and provide a plurality of update packages 110 ... the update generator 102 may create a version manifest (reference software), which comprises a list of archived update packages 110 ...; P. 15, 1st full Para - ... a server manifest comprising a list of archived update packages ... may transfer the server manifest to the one or more client devices ... then review the manifest and submit a request for the update package ... to be transferred from the update server array 122 to the one or more client devices ...);

Further, O'Neil discloses the system and methods which provide a developer with an improved ability to create update information as needed and additionally allow users to proceed through a simplified update path which is not error-prone and can be performed more quickly than through the use of existing technologies (e.g., P. 4, last 2nd full Para); further, the update installation process may recognize the presence of multiple available updates (versions) and proceed to update the file or component using the most current version available without sequentially applying each possible update package (e.g., P. 22, 2nd full Para) but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Method and System for Distribution of File Updates*, Crudele discloses the following:

- generating a third update package for updating the at least one software application, the third update package being generated based upon difference information between the first and second update packages (e.g., [0025] - ... The

software package file comprises the delta files ... starting installation of the new code with the delta files ...; [0035] ... This sequence of steps is repeated until the end of delta file is reached ... for each of the delta files contained in the software package)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Crudele into the O'Neil's system to further provide other limitations stated above in the O'Neil system.

The motivation is that it would further enhance the O'Neil's system by taking, advancing and/or incorporating the Crudele's system which provides one major advantage of the current solution is that it applies at a byte level, this means that the method is applicable not only to the distribution of applications but also to the distribution of data files as once suggested by Crudele (e.g., [0016])

Furthermore, O'Neil discloses updating the at least one software application using the third update package (e.g., P. 22, 1st full Para - ... proceed to update the file or component using the most current version available without sequentially applying each possible update package (i.e. version 1 updated directly to version 3) ...)

25. **As to claim 22** (Original), O'Neil discloses a method for updating software in an electronic device, the method comprising:

- generating a first update package for updating at least one software application, the first update package being generated based upon difference information between a first software version and a reference software corresponding to the at

least one software application (e.g., Fig. 1A, elements 110 – Update Package; 102 – Update Generator; 106 – First Version; 108 – Second Version; P. 5, 1st non-full Para - ... identifies differences between of the updated operating code and the resident operating code and thereafter generates an update package ...; P. 5, last Para - ... an update generator that performs a version comparison between the first code version and the second code version to identify pattern differences ...; P. 12, 1st non-full Para - ... Alternatively, the update generator 102 may be equipped with the ability to generate and provide a plurality of update packages 110 ... the update generator 102 may create a version manifest (reference software), which comprises a list of archived update packages 110 ...; P. 15, 1st full Para - ... a server manifest comprising a list of archived update packages ... may transfer the server manifest to the one or more client devices ... then review the manifest and submit a request for the update package ... to be transferred from the update server array 122 to the one or more client devices ...);

- generating a second update package for updating the at least one software application, the second update package being generated based upon difference information a second software version and the reference software corresponding to the at least one software application (e.g., Fig. 1A, elements 110 – Update Package; 102 – Update Generator; 106 – First Version; 108 – Second Version; P. 5, 1st non-full Para - ... identifies differences between of the updated operating code and the resident operating code and thereafter generates an

update package ...; P. 5, last Para - ... an update generator that performs a version comparison between the first code version and the second code version to identify pattern differences ...; P. 12, 1st non-full Para - ... Alternatively, the update generator 102 may be equipped with the ability to generate and provide a plurality of update packages 110 ... the update generator 102 may create a version manifest (reference software), which comprises a list of archived update packages 110 ...; P. 15, 1st full Para - ... a server manifest comprising a list of archived update packages ... may transfer the server manifest to the one or more client devices ... then review the manifest and submit a request for the update package ... to be transferred from the update server array 122 to the one or more client devices ...);

Further, O'Neil discloses the system and methods which provide a developer with an improved ability to create update information as needed and additionally allow users to proceed through a simplified update path which is not error-prone and can be performed more quickly than through the use of existing technologies (e.g., P. 4, last 2nd full Para); further, the update installation process may recognize the presence of multiple available updates (versions) and proceed to update the file or component using the most current version available without sequentially applying each possible update package (e.g., P. 22, 2nd full Para) but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Method and System for Distribution of File Updates*, Crudele discloses:

- generating a third update package for updating the at least one software application, the third update package being generated based upon difference information between the first and second update packages (e.g., [0025] - ... The software package file comprises the delta files ... starting installation of the new code with the delta files ...; [0035] ... This sequence of steps is repeated until the end of delta file is reached ... for each of the delta files contained in the software package)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Crudele into the O'Neil's system to further provide other limitations stated above in the O'Neil system.

The motivation is that it would further enhance the O'Neil's system by taking, advancing and/or incorporating the Crudele's system which provides one major advantage of the current solution is that it applies at a byte level, this means that the method is applicable not only to the distribution of applications but also to the distribution of data files as once suggested by Crudele (e.g., [0016])

Furthermore, O'Neil discloses updating the at least one software application using the third update package (e.g., P. 22, 1st full Para - ... proceed to update the file or component using the most current version available without sequentially applying each possible update package (i.e. version 1 updated directly to version 3) ...)

26. **As to claim 23** (Original), O'Neil discloses a system for updating software, the system comprising:

- an electronic device capable of having software installed thereon (e.g., P. 17, 2nd full Para - ... the update management system may be used .. such as cellular or mobile phones ...);
- a first update package generator for generating update packages based upon difference information between different versions of software (e.g., Fig. 1A, elements 110 – Update Package; 102 – Update Generator; 106 – First Version; 108 – Second Version; P. 5, 1st non-full Para - ... identifies differences between of the updated operating code and the resident operating code and thereafter generates an update package ...; P. 5, last Para - ... an update generator that performs a version comparison between the first code version and the second code version to identify pattern differences ...; P. 12, 1st non-full Para - ... Alternatively, the update generator 102 may be equipped with the ability to generate and provide a plurality of update packages 110 ... the update generator 102 may create a version manifest (reference software), which comprises a list of archived update packages 110 ...; P. 15, 1st full Para - ... a server manifest comprising a list of archived update packages ... may transfer the server manifest to the one or more client devices ... then review the manifest and submit a request for the update package ... to be transferred from the update server array 122 to the one or more client devices ...);

Further, O'Neil discloses the system and methods which provide a developer with an improved ability to create update information as needed and additionally allow users to proceed through a simplified update path which is not error-prone and can performed

more quickly than through the use of existing technologies (e.g., P. 4, last 2nd full Para); further, the update installation process may recognize the presence of multiple available updates (versions) and proceed to update the file or component using the most current version available without sequentially applying each possible update package (e.g., P 22, 2nd full Para) but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Method and System for Distribution of File Updates*, Crudele discloses the following:

- a second update package generator for generating update packages based upon difference information between different update packages (e.g., [0025] - ... The software package file comprises the delta files ... starting installation of the new code with the delta files ...; [0035] ... This sequence of steps is repeated until the end of delta file is reached ... for each of the delta files contained in the software package)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Crudele into the O'Neil's system to further provide other limitations stated above in the O'Neil system.

The motivation is that it would further enhance the O'Neil's system by taking, advancing and/or incorporating the Crudele's system which provides one major advantage of the current solution is that is applies at a byte level, this means that the method is applicable not only to the distribution of applications but also to the distribution of data files as once suggested by Crudele (e.g., [0016])

Furthermore, O'Neil discloses a software delivery device for delivering at least one update package generated based upon difference information between different update packages to the electronic device (e.g., P. 22, 1st full Para - ... proceed to update the file or component using the most current version available without sequentially applying each possible update package (i.e. version 1 updated directly to version 3) ...)

27. **As to claim 24** (Original), O'Neil discloses a system for updating software, the system comprising:

- an electronic device capable of having software installed thereon (e.g., P. 17, 2nd full Para - ... the update management system may be used .. such as cellular or mobile phones ...);
- a first update package generator for generating update packages based upon difference information between a version of software and a reference software corresponding to at least one software application (e.g., Fig. 1A, elements 110 – Update Package; 102 – Update Generator; 106 – First Version; 108 – Second Version; P. 5, 1st non-full Para - ... identifies differences between of the updated operating code and the resident operating code and thereafter generates an update package ...; P. 5, last Para - ... an update generator that performs a version comparison between the first code version and the second code version to identify pattern differences ...; P. 12, 1st non-full Para - ... Alternatively, the update generator 102 may be equipped with the ability to generate and provide a plurality of update packages 110 ... the update generator 102 may create a

version manifest (reference software), which comprises a list of archived update packages 110 ...; P. 15, 1st full Para - ... a server manifest comprising a list of archived update packages ... may transfer the server manifest to the one or more client devices ... then review the manifest and submit a request for the update package ... to be transferred from the update server array 122 to the one or more client devices ...);

Further, O'Neil discloses the system and methods which provide a developer with an improved ability to create update information as needed and additionally allow users to proceed through a simplified update path which is not error-prone and can performed more quickly than through the use of existing technologies (e.g., P. 4, last 2nd full Para); further, the update installation process may recognize the presence of multiple available updates (versions) and proceed to update the file or component using the most current version available without sequentially applying each possible update package (e.g., P 22, 2nd full Para) but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Method and System for Distribution of File Updates*, Crudele discloses the following:

- a second update package generator for generating update packages based upon difference information between different update packages (e.g., [0025] - ... The software package file comprises the delta files ... starting installation of the new code with the delta files ...; [0035] ... This sequence of steps is repeated until the end of delta file is reached ... for each of the delta files contained in the software package)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Crudele into the O'Neil's system to further provide other limitations stated above in the O'Neil system.

The motivation is that it would further enhance the O'Neil's system by taking, advancing and/or incorporating the Crudele's system which provides one major advantage of the current solution is that it applies at a byte level, this means that the method is applicable not only to the distribution of applications but also to the distribution of data files as once suggested by Crudele (e.g., [0016])

Furthermore, O'Neil discloses a software delivery device for delivering at least one update package generated based upon difference information between different update packages to the electronic device (e.g., P. 22, 1st full Para - ... proceed to update the file or component using the most current version available without sequentially applying each possible update package (i.e. version 1 updated directly to version 3) ...)

Conclusion

28. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ben C. Wang whose telephone number is 571-270-1240. The examiner can normally be reached on Monday - Friday, 8:00 a.m. - 5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Ben C Wang/
Ben C. Wang
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192